A Context-Based View of Deep Neural Networks

Most discussions of *context* in deep learning focus on Transformer attention. However, a more general notion of context may be useful.

This note proposes a simple conceptual viewpoint.

Context is broader than attention

- A context is simply information that determines **which computation is performed**.
- Attention is one way of creating context.
- ReLU activation patterns also create context.
- More generally, any routing or gating mechanism creates context.

ReLU as parameter selection

A ReLU network is usually described as activating or deactivating neurons.

An equivalent viewpoint is:

- Every input activates a different subset of parameters.
- Every input therefore selects a different linear mapping.
- The network behaves as a large collection of context-dependent linear models.

This parameter-selection interpretation appears much less frequently than the traditional activation-function explanation, despite being mathematically equivalent.

A layer-centric interpretation

Each layer can be viewed as performing the following computation.

- Receive an input representation.
- Determine its context.
- Select an appropriate linear mapping.
- Produce the next representation.

The layer therefore acts as a context-conditioned linear operator.

What is learned?

A useful intuition is that each context-specific linear mapping is learning:

> **The best average linear mapping for that layer and that context.**

Not the exact mapping.

Not stored examples.

But the average transformation that has historically produced the lowest training error whenever that context occurs.

Fractional associative memories

This suggests another interpretation.

Each context-conditioned linear mapping is a **fractional associative memory**.

Rather than storing complete concepts, it stores a small piece of statistical knowledge.

Conceptually it remembers:

> "When representations entering this layer look like this, this linear transformation has usually worked well."

Composition

The power of deep networks comes from composition.

- Every layer learns its own collection of average context-dependent mappings.
- The output of one layer becomes the input to the next.
- Backpropagation couples the learning of all layers so that these local mappings become mutually compatible.

The overall nonlinear computation emerges from composing many locally learned average linear mappings rather than from any single layer performing the entire task.

A possible perspective

Viewed this way, a deep network becomes a hierarchy of context-conditioned fractional associative memories.

Each layer contributes local statistical transformations.

Backpropagation coordinates these transformations.

The resulting composition approximates the desired global nonlinear function.